Performance assessment and design of finite length LDPC codes for the Gaussian wiretap channel

Marco Baldi, Giacomo Ricciutelli, Nicola Maturo, Franco Chiaraluce,
DII, Università Politecnica delle Marche,
Ancona, Italy
Email: {m.baldi, n.maturo, f.chiaraluce}@univpm.it, g.ricciutelli@pm.univpm.it

***Abstract*—In this work we study the reliability and secrecy performance achievable by practical low-density parity-check (LDPC) codes over the Gaussian wiretap channel. While several works have already addressed this problem in asymptotic conditions, i.e., under the hypothesis of codewords of infinite length, only a few approaches exist for the finite length regime. We propose an approach to measure the performance of practical codes and compare it with that achievable in asymptotic conditions. Moreover, based on the secrecy metrics we adopt to achieve this target, we propose a code optimization algorithm which allows to design irregular LDPC codes able to approach the ultimate performance limits even at moderately small codeword lengths (in the order of 10000 bits).**

I. Introduction

Coding for the Gaussian wiretap channel is a well-established research topic, but there are some partially unsolved and challenging problems. One of these problems is to study the secrecy performance in the finite code length regime, and to design optimized finite length codes. One of the most common metrics to assess the performance of finite length codes used for transmissions is the average bit error rate (BER) achieved by using some (possibly optimal) decoder. On the other hand, the secrecy performance over wiretap channels is classically measured using information-theoretic metrics, like the secrecy capacity, and in asymptotic conditions (e.g., infinite code length and random coding).

For example, in [1], the authors consider a discrete memoryless channel (DMC) (that is, a binary erasure channel (BEC) or binary symmetric channel (BSC)) model for both the main and wiretapper's channels, and design optimized regular LDPC codes for these channels. They show that their approach achieves the secrecy capacity when the wiretap channel consists of symmetric DMCs. No continuous channels are considered, and the secrecy capacity is achieved in the asymptotic regime (i.e., with infinite length codes).

The BER as a secrecy metric has instead been used in [2], where a coding scheme able to achieve a BER very close to 0.5 for the eavesdropper and very low for the authorized channel is proposed. In [2], the authors use differential evolution to design optimized LDPC codes able to achieve the desired BER targets while keeping the quality ratio between the main and the eavesdropper's channels (named security gap) as small as possible. The proposed coding scheme is based on puncturing and, thanks to the BER-based analysis, is applicable at finite block lengths. A similar solution, but without the need of puncturing, has been proposed in [3], and extended in [4] to the case of parallel channels.

A bridge between information theoretic and error rate-based secrecy measures is presented in [5], where however the main goal is to propose a secret key sharing scheme for the wiretap channel, and the presence of an error-free public channel between the source and destination is considered, which helps the secret sharing process. By using regular LDPC codes, the authors show that the key capacity can be achieved in the asymptotic regime. Irregular LDPC codes are instead considered for the finite code length regime, and a density evolution based linear program is used to design them. The same approach is followed in [6] to assess the performance of punctured LDPC codes over the Gaussian wiretap channel.

Inspired by such works, in this paper we study the performance of finite length LDPC codes over the Gaussian wiretap channel, by defining suitable metrics to assess how far they are from optimality, which is achieved in asymptotic conditions. This permits us to explore the capacity-equivocation regions of these codes in the finite length regime, and without using puncturing. We also propose a twofold code optimization tool which allows to design optimal codes in terms of the considered metrics. Similar twofold code optimizations have been proposed for the relay channel [7]–[9], but no solution has been presented for the wiretap channel, at our best knowledge. We show that our approach allows to achieve great flexibility in the choice of the system parameters, as well as higher security levels with respect to previous solutions based on punctured LDPC codes [6].

The organization of the paper is as follows. In Section II we present the system model and the metrics we use to assess performance in asymptotic and finite length conditions. In Section III we describe the code design requirements. In Section IV we propose our code optimization approach. In Section V we provide and discuss some numerical results. Finally, Section VI concludes the paper.

II. System model and metrics

We consider the classical Gaussian wiretap channel model, in which a sender, named Alice, transmits a secret message $\mathcal{M}$. She encodes her message into the n-symbol codeword X^n, which uniquely depends on $\mathcal{M}$ and on some random message

This work was supported in part by the MIUR project "ESCAPADE" (Grant RBFR105NLC) under the "FIRB – Futuro in Ricerca 2010" funding program.

$\mathcal{R}$ generated by Alice. We consider binary coding, therefore X^n actually is an n-bit codeword. If the secret message is k_s bits long and the random message is k_r bits long, the code rate is $R_c = (k_s + k_r)/n = k/n$. The secret message rate, instead, is $R_s = k_s/n$.

Transmission occurs over a Gaussian channel for both the authorized receiver, named Bob, and the eavesdropper, named Eve. The noisy codewords received by Bob and Eve are denoted by Y^n and Z^n, respectively. In order to achieve successful transmission of $\mathcal{M}$ over this channel, both the following targets must be fulfilled:

i) $\mathcal{M}$ must be reliably decoded by Bob, i.e., with a sufficiently small error rate (*reliability target*),
ii) the information about $\mathcal{M}$ gathered by Eve must be sufficiently small (*security target*).

Concerning the reliability target, in ideal conditions (i.e., infinite code length and random coding) the channel capacity can be used as the ultimate code rate limit. In the finite length regime, instead, a practical code must be designed to allow Bob to achieve a sufficiently low error rate in decoding the secret message. Concerning the security target, some classical information theoretic secrecy metrics are only useful in the asymptotic regime. In fact, denoting by $\mathrm{I}(x;y)$ the mutual information between x and y, we have [10]:

- Strong secrecy when the total amount of information leaked about $\mathcal{M}$ through observing Z^n goes to zero as n goes to infinity, i.e., $\lim_{n\to\infty} \mathrm{I}(\mathcal{M}; Z^n) = 0$.
- Weak secrecy when the rate of information leaked about $\mathcal{M}$ through observing Z^n goes to zero as n goes to infinity, i.e., $\lim_{n\to\infty} \mathrm{I}(\mathcal{M}; Z^n)/n = 0$.

So, these metrics are not useful in order to assess the performance in finite length conditions and compare it with that in the asymptotic regime.

However, another metric can be exploited, which was already used in Wyner's original work [11]. According to [11], transmission is accomplished in perfect secrecy when the wiretapper equivocation rate on the secret message, $R_e = \frac{1}{n}\mathrm{H}(\mathcal{M}|Z^n)$, with $\mathrm{H}(\cdot)$ denoting the entropy function, equals the entropy of the data source. We consider independent and identically distributed secret messages, therefore the source entropy rate is equal to R_s. So, perfect secrecy is achieved when the equivocation rate R_e equals the secret message rate R_s, i.e.,

$$\widetilde{R_e} = R_e/R_s = 1. \tag{1}$$

$\widetilde{R_e}$ is called fractional equivocation rate.

Actually, the ultimate limit achievable by the equivocation rate is the secrecy capacity $C_s = C_B - C_E$, where C_B and C_E are Bob's and Eve's channel capacities, respectively. For a binary-input channel with additive white Gaussian noise (AWGN) and signal-to-noise ratio (SNR) γ, the capacity is given by the following expression:

$$C(\gamma) = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{(y-\sqrt{\gamma})^2}{2}} \log_2\left(1 + e^{-2y\sqrt{\gamma}}\right) dy. \tag{2}$$

Then, the target is to maximize R_e in such a way as to approach the secrecy capacity. On the other hand, when considering finite length codes, it is expected that $C_s < R_s$ and another valuable issue is the evaluation of the gap between the secret message rate and the secrecy capacity. Numerical examples will be presented in Section V.

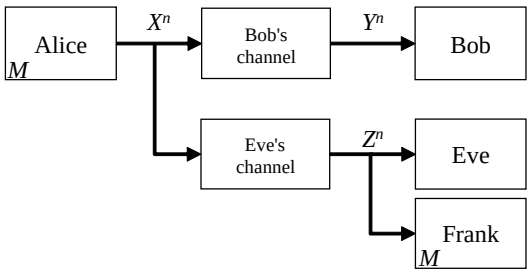

Fig. 1. Wiretap channel model employed in the study.

Concerning the computation of the equivocation rate, it can be shown that [6]:

$$\begin{aligned} R_e = \frac{1}{n} \big[&\mathrm{H}(X^n) - \mathrm{I}(X^n; Z^n) + \mathrm{H}(\mathcal{M}|Z^n, X^n) \\ &-\mathrm{H}(X^n|\mathcal{M}, Z^n)\big]. \end{aligned} \tag{3}$$

From (3) it results that this formulation of Eve's equivocation rate requires to compute the quantity $\mathrm{H}(X^n|\mathcal{M}, Z^n)$, that is, the entropy of X^n conditioned to receiving Z^n and knowing the secret message $\mathcal{M}$. Eve obviously does not know the secret message, therefore we suppose the existence of another (fictitious) receiver in the same position as Eve's, knowing the secret message $\mathcal{M}$. We denote such a receiver as Frank: he receives the same vector Z^n as Eve but, differently from Eve, he has perfect knowledge of the secret message $\mathcal{M}$. Then, he tries to decode Z^n for recovering the random message $\mathcal{R}$, which is the only source of uncertainty for Frank in order to reconstruct X^n. The resulting wiretap channel model is schematically depicted in Fig. 1. The letter M inside Alice's and Frank's boxes points out that the message is known to both Alice and Frank.

Let us suppose that, in these conditions, Frank experiences a decoding error probability (or codeword error rate (CER)) equal to η. By Fano inequality we have $\mathrm{H}(X^n|\mathcal{M}, Z^n) \leq 1 + k_r\eta$. We also have $\mathrm{H}(X^n) = k$ and $\mathrm{H}(\mathcal{M}|Z^n, X^n) \leq \mathrm{H}(\mathcal{M}|X^n) = 0$. Concerning Eve's channel mutual information $\mathrm{I}(X^n; Z^n)$, we could obtain a tight upper bound on it as proposed in [12], by taking into account the code length and the target error rate. However, by using the classical bound $\mathrm{I}(X^n; Z^n) \leq nC_E$, we obtain a limit value which is independent of Eve's error rate. Such a value cannot be overcome even if Eve's error rate changes, therefore it represents a conservative choice for our purposes. Based on these considerations, we can find a lower bound on Eve's equivocation rate about the secret message as [6]:

$$\begin{aligned} R_e &\geq \frac{1}{n}\left[k - nC_E - k_r\eta - 1\right] = \\ &= R_c - C_E - (R_c - R_s)\eta - \frac{1}{n} = R_e^*. \end{aligned} \tag{4}$$

By looking at (4), it is evident that this metric is well suited to assess the secrecy performance of practical, finite

length codes. In fact, the code length is taken into account, and the error rate experienced by Frank can be estimated for practical codes through numerical simulations. Its value obviously depends on Frank's SNR, which is the same as Eve's, and therefore, according to (2), it determines C_E. It follows that, for a fixed code length and rate, the equivocation rate can be maximized by optimizing the choice of the pair (η, C_E).

III. Code design

An LDPC code with rate $R_c = k/n$ is defined through its parity-check matrix $\mathbf{H}$ of size $(n-k) \times n$. Alternatively, the LDPC code can be represented through a Tanner graph, that is a bipartite graph composed of variable and check nodes, which correspond to the codeword bits and the parity-check equations, respectively. Noting by h_{ij} the (i,j)-th element of $\mathbf{H}$, there is an edge between the j-th variable node and the i-th check node iff $h_{ij} = 1$. The number of edges connected to a node is called degree of that node. The following two polynomials are commonly used to denote the variable and check node degree distributions:

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}, \qquad \rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1} \tag{5}$$

where d_v and d_c are the maximum variable and check node degrees, respectively. In $\lambda(x)$ ($\rho(x)$), the coefficient λ_i (ρ_j) coincides with the fraction of edges connected to the variable (check) nodes having degree i (j). Therefore, $\lambda(x)$ and $\rho(x)$ are defined from the edge perspective. The code rate can be expressed as:

$$R_c = 1 - \frac{\sum_{i=2}^{d_v} \rho_i / i}{\sum_{j=2}^{d_c} \lambda_j / j}. \tag{6}$$

The most common LDPC code decoding algorithm, which is an instance of the well-known belief propagation principle, is based on the exchange of soft messages about each received bit between the nodes of its Tanner graph. Therefore, the performance of an LDPC code depends on the connections among the nodes of its Tanner graph. Indeed, a variable node with a greater number of connected edges has more parity-check equations which verify its associated bit. On the other hand, check nodes with low degrees correspond to parity-check equations with less unknowns. The optimization of the code performance under message passing decoding consists in finding the best tradeoff between these two effects, and this usually requires irregular degree distributions. The well-known density evolution algorithm, proposed in [13], aims at optimizing the pair $(\lambda(x), \rho(x))$ based on the statistics of the decoder messages. However, differently from classical transmission problems, in our setting the same code (chosen by Alice) is used by three receivers: Bob, Eve and Frank, and the code optimization should take this into account.

Let us consider a systematic encoder and let us suppose that the transmitted codeword is $\mathbf{c} = [\mathcal{M}|\mathcal{R}|\mathcal{P}]$, where $\mathcal{M}$ is the k_s-bit secret message, $\mathcal{R}$ is the k_r-bit random message and $\mathcal{P}$ is the r-bit redundancy vector added by the encoder.

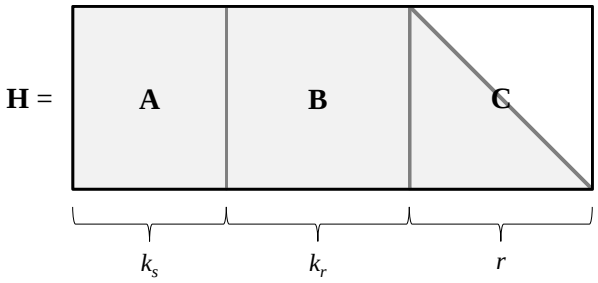

Fig. 2. Parity-check matrix of the considered codes.

Obviously, systematic encoding shall be avoided in security applications, especially if source coding is not optimal. In fact, in such a case, Eve could look at the systematic part of the received codeword and gather some information about the secret message parts which are less affected by errors. In practical systems, systematic encoding can be easily avoided by scrambling the information bits prior to encoding [3]. Having this clearly in mind, for our code design and analysis purposes it is convenient to keep the assumption of systematic encoding. Under this hypothesis, the code parity-check matrix can be divided into three blocks as shown in Fig. 2, corresponding to the three parts of the codeword $\mathbf{c}$. Bob must use the whole matrix to decode for both the secret and random messages (since he does not know in advance any of them), although in the end he is interested only in $\mathcal{M}$. Eve is in the same condition, although she receives the signal through a different channel. Frank, instead, has perfect knowledge of $\mathcal{M}$, and only needs to decode for $\mathcal{R}$. Therefore, he can precompute $\mathbf{A} \cdot \mathcal{M}^T = \mathbf{s}$, were ^{T} denotes transposition. Then, he can use $\mathbf{s}$ as a syndrome vector and focus on the reduced parity-check system:

$$[\mathbf{B}|\mathbf{C}] \cdot [\mathcal{R}|\mathcal{P}]^T = \mathbf{H}' \cdot \mathbf{c}'^T = \mathbf{s}.$$

Obviously, decoding for a vector having an all-zero syndrome or a different syndrome is equivalent, due to the code linearity. Hence, Frank performs decoding through the LDPC code defined by $\mathbf{H}'$, having rate $R_F = k_r/(k_r + r)$. The code rate for Bob instead coincides with the overall code rate, i.e., $R_B = k/n$. It follows that $R_F = \frac{R_B - R_s}{1 - R_s}$. In the setting we consider, it is important that both Bob's and Frank's codes are optimized. In fact, an optimized code for Bob allows to approach the channel capacity, which is the ultimate limit for the reliability target. An optimized code for Frank instead serves to achieve the desired η with the smallest possible SNR. Since Frank's SNR is the same as Eve's, this reduces Eve's channel capacity C_E.

IV. Code optimization

We propose an optimization strategy for Bob's and Frank's codes based on the density evolution algorithm, which is commonly used to optimize a single code, with some modifications in order to consider the joint optimization target. In Section IV-A we briefly recall the steps of the single code optimization and then in Section IV-B we describe our strategy for the joint code optimization. In this work, as in [13], we use the

density evolution algorithm with Gaussian approximation of the decoder messages.

A. Single optimization

The density evolution algorithm is well-known in the literature; therefore, for the sake of brevity, we report here only the main equations of [13], as they are used in the proposed joint code optimization.

Given $\rho(x)$, R_c and d_v, the optimization of $\lambda(x)$ of a single code is possible by applying the following constraints:

C_1 - Rate constraint:

$$\sum_{i=2}^{d_v} \frac{\lambda_i}{i} = \frac{1}{1-R_c} \sum_{i=2}^{d_c} \frac{\rho_i}{i}. \tag{7}$$

C_2 - Proportion distribution constraint:

$$\sum_{i=2}^{d_v} \lambda_i = 1. \tag{8}$$

C_3 - Convergence constraint (from [13, Eq. (16)]):

$$r > h(s,r), \quad \forall r \in (0, \phi(s)) \tag{9}$$

where $s = \frac{2}{\sigma^2}$, σ^2 being the noise variance, and $\phi(\cdot)$ will be defined in (12). For $0 < s < \infty$ and $0 < r \leq 1$, we define $h(s,r)$ in (9) as follows:

$$h(s,r) = \sum_{i=2}^{d_v} \lambda_i h_i(s,r) \tag{10}$$

where

$$h_i(s,r) = \phi\left(s + (i-1)\sum_{j=2}^{d_c} \rho_j \phi^{-1}\left(1-(1-r)^{j-1}\right)\right). \tag{11}$$

In (9) and (11),

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{+\infty} \tanh\frac{u}{2} e^{-\frac{(u-x)^2}{4x}} du, & \text{if } x > 0 \\ 1, & \text{if } x = 0. \end{cases} \tag{12}$$

Condition (9) is equivalent to impose that $r_l(s) \to 0$ for $l \to \infty$ [13], with $r_l = h(s, r_{l-1})$ and $r_0 = \phi(s)$.

C_4 - Stability condition:

$$\lambda_2 < \frac{e^{\frac{1}{2\sigma^2}}}{\sum_{j=2}^{d_c} \rho_j (j-1)}. \tag{13}$$

In the single code optimization, the code threshold s^* is defined as the minimum s for which the constraints $[C_1 - C_4]$ are satisfied. From the definition of s, it is evident that s^* corresponds to the maximum noise variance σ^2 for which the constraints are verified.

B. Joint optimization

In order to perform the joint optimization of Bob's and Frank's codes, we must impose that Frank's code is somehow *contained* in Bob's code (in other terms, that Frank's parity-check matrix is a sub-matrix of Bob's parity-check matrix). Therefore, in addition to the constraints in Section IV-A, we need another condition. To obtain this further constraint, we introduce the polynomial $\tilde{\lambda}(x)$ which corresponds to the node perspective of $\lambda(x)$. In $\tilde{\lambda}(x)$, the fraction of nodes of degree i can be derived from $\lambda(x)$ through the following formula:

$$\tilde{\lambda}_i = \frac{\lambda_i / i}{\sum_{k=2}^{d_v} \lambda_k / k}. \tag{14}$$

In order to obtain the check node degree distributions from the node perspective $\tilde{\rho}(x)$, a similar formula can be applied to the check nodes degree distributions from the edge perspective. This can be easily achieved by replacing in (14) $\lambda(x)$ with $\rho(x)$, $\tilde{\lambda}(x)$ with $\tilde{\rho}(x)$ and d_v with d_c.

Since Bob's parity-check matrix contains Frank's parity-check matrix, the number of variable nodes in Bob's Tanner graph having some fixed degree must be greater than or equal to that of variable nodes in Frank's Tanner graph having the same degree. Hence, we must take into account the following further constraint:

C_5 - Joint optimization constraint:

$$\tilde{\lambda}_{B,i} \geq \tilde{\lambda}_{F,i}, \quad \forall i \in \left[2, 3, 4, \ldots, d_v^{(F)}\right], \tag{15}$$

where $\tilde{\lambda}_B(x)$ and $\tilde{\lambda}_F(x)$ are Bob's and Frank's variable node degree distributions from the node perspective, respectively, and $d_v^{(F)}$ is Frank's maximum variable node degree. C_5 adds to $[C_1 - C_4]$ and the optimum $\lambda_B(x)$ must satisfy all these constraints.

In the joint optimization algorithm, we define the convergence threshold as the maximum of $c = \sigma_B^2 + \sigma_F^2$, denoted by c^*, for which the constraints $[C_1 - C_5]$ are satisfied. In the expression of c, σ_B^2 and σ_F^2 are Bob's and Frank's noise variances, respectively. It should be noted that this procedure differs from optimizing the two codes separately. In fact, in principle, we could first optimize Frank's code, and then try to optimize Bob's code by taking account the degree distributions obtained for Frank and the constraint C_5. This, however, could impose too strong constraints on Bob's code degree distribution, thus preventing to find a good solution for him, too. In fact, some solutions may exist for which neither Bob's nor Frank's degree distributions are individually optimal, but their joint performance is optimal.

As in [13], in order to design the check node degree distribution, we adopt a concentrated distribution (i.e., with only two degrees, concentrated around the mean). It is widely recognized that this solution, though very simple, is able to achieve very good performance. Hence, for each pair $(\lambda_F(x), \lambda_B(x))$, we obtain the pair $(\tilde{\rho}_F(x), \tilde{\rho}_B(x))$ by using the following formula, valid for both Bob and Frank:

$$\tilde{\rho}(x) = a x^{\lfloor c_m \rfloor} + b x^{\lceil c_m \rceil}, \tag{16}$$

where $c_m = \frac{E}{r} = \frac{\sum_j \tilde{\lambda}_j \cdot j}{(1-R_c)}$ and E is the total number of edges in the Tanner graph. The values a and b are computed as

$$a = \lceil c_m \rceil - c_m, \qquad b = c_m - \lfloor c_m \rfloor. \tag{17}$$

In (16) and (17), $\lceil c_m \rceil$ and $\lfloor c_m \rfloor$ represent the ceiling and floor value of c_m, respectively.

V. Numerical results

In order to provide some practical examples, we use the procedure described in Section IV-B to design several codes with $d_v^{(B)} = d_v^{(F)} = 50$. We consider code rates $R_c = R_B = 0.35, 0.5, 0.75$ and several values of $R_s < R_B$. The degree distributions obtained through the joint optimization procedure are reported in Table I. Concerning the choice of the degrees of x allowed in the two polynomials, the only constraints we impose are that they must not overcome the maximum values $d_v^{(B)}$ and $d_v^{(F)}$, and that the number of nodes of degree 2 must be such that the stability condition (13) is met by both codes.

To provide some examples of finite length codes, we consider LDPC codes with length $n_1 = 10000$ and $n_2 = 50000$; Frank's code length is then obtained from these values by considering the submatrix $\mathbf{H}'$. Once having defined the degree distributions, the parity-check matrices are designed through the Progressive Edge Growth (PEG) algorithm [14]. The numerical results are obtained by considering, for all coding schemes, binary phase shift keying (BPSK) modulation over the AWGN channel. When considering finite length codes, through numerical simulations we are able to determine the values of the SNR per bit (E_b/N_0) that ensure a given CER. These values are reported in Table II, for both Bob and Frank, assuming CER $= 10^{-2}$ and several values of R_s. In the table, the values of $\left.\frac{E_b}{N_0}\right|_{th}$ identify the codes convergence thresholds obtained through density evolution. These values represent the ultimate performance bounds achievable in asymptotic conditions (i.e., infinite code length). The values of $\left.\frac{E_b}{N_0}\right|_{n_1}$ and $\left.\frac{E_b}{N_0}\right|_{n_2}$ instead represent the SNR working points, estimated through simulations, for the practical codes with lengths n_1 and n_2, respectively. We observe from Table II that, for Bob's code, the finite length performance approaches the asymptotic threshold as the code rate increases. Indeed, for $R_B = 0.75$ and code length equal to n_1 and n_2, the gap between the asymptotic threshold and the finite length codes performance is about 0.4 dB and 0.2 dB, respectively.

As a security metric we use the lower bound R_e^* on the equivocation rate, computed according to (4) and the values in Table II. The secrecy capacity C_s, that represents the ultimate limit achievable by the equivocation rate, is also computed for the cases of interest, and used as a benchmark. We compute C_s under the hypothesis of ideal coding, i.e., that Bob's and Frank's code rates coincide with the respective channel capacities. Since Frank's and Eve's channels coincide, it follows that $C_s = R_B - R_F = R_s \frac{1-R_B}{1-R_s}$. In order to assess if practical codes can approach the perfect secrecy condition (1), we then compute the fractional lower bound on the equivocation rate $\widetilde{R_e^*} = R_e^*/R_s$ both in asymptotic conditions and in the finite code length regime, and compare its values with the fractional secrecy capacity $\widetilde{C_s} = C_s/R_s = \frac{1-R_B}{1-R_s}$. The values so obtained are reported in Fig. 3, for the same values of R_s considered in Tables I and II. As an example, for the considered code parameters and $R_s = 0.725$, we find that in asymptotic conditions the designed codes approach the secrecy capacity and the perfect secrecy condition. Notably, even using relatively short codes, with 10000-bit codewords, the fractional equivocation rate is close to 0.8. For the sake of comparison, we consider some results reported in [6] for the scheme based on punctured LDPC codes. The corresponding points are marked with an asterisk in Fig. 3. Those results consider codes with length $n = 10^6$, at which the performance of LDPC codes usually approaches the density evolution threshold. However, the asymptotic performance achieved by the degree distributions found through the proposed approach exhibits some gain at the same secret message rates. Furthermore, for $R_s = 0.43$, even our schemes with $n = 10000$ and $n = 50000$ outperform that proposed in [6] with $n = 10^6$.

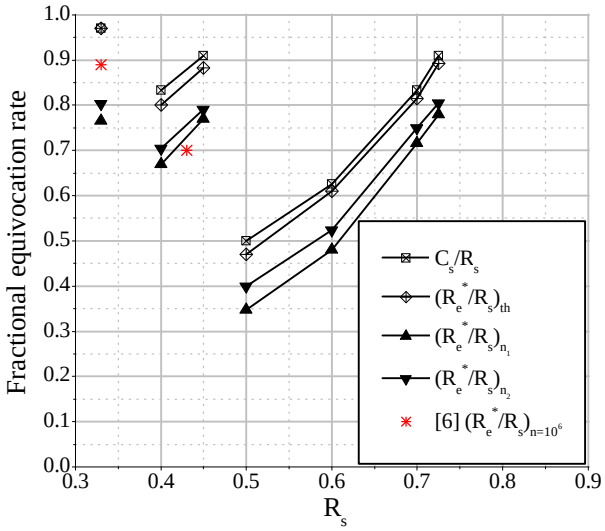

Fig. 3. Comparison between $\frac{C_s}{R_s}$, $\frac{R_e^*}{R_s}$ calculated through the asymptotic threshold values, $\frac{R_e^*}{R_s}$ for code length n_1, and $\frac{R_e^*}{R_s}$ for code length n_2, as a function of R_s.

From Fig. 3 it results that the best performance in terms of Eve's equivocation rate is achieved when the secret message rate approaches the code rate. This could seem counterintuitive, since suggests to use few random bits to confuse the eavesdropper. However, in this condition R_F is small and Frank is able to reach the desired performance at low SNR. The latter coincides with Eve's channel SNR, therefore Eve's equivocation rate is large. On the other hand, imposing that Eve's channel has a too low SNR is not realistic, therefore some randomness shall always be used in order to relax the constraints on Eve's channel quality.

VI. Conclusion

We have studied the performance of practical LDPC coded transmissions over the Gaussian wiretap channel. By using suitable reliability and security metrics, we have computed performance bounds in the asymptotic regime and assessed the achievable performance under the hypothesis of finite

TABLE I

DEGREES DISTRIBUTION PAIRS OBTAINED WITH THE TECHNIQUE DESCRIBED IN SECTION IV-B FOR SEVERAL VALUES OF R_s AND R_B.

R_s	0.33		0.4		0.45		0.5		0.6		0.7		0.725	
R_B	0.35		0.5		0.5		0.75		0.75		0.75		0.75	
i	$\lambda_{F,i}$	$\lambda_{B,i}$	$\lambda_{F,i}$	$\lambda_{B,i}$	$\lambda_{F,i}$	$\lambda_{B,i}$	$\lambda_{F,i}$	$\lambda_{B,i}$	$\lambda_{F,i}$	$\lambda_{B,i}$	$\lambda_{F,i}$	$\lambda_{B,i}$	$\lambda_{F,i}$	$\lambda_{B,i}$
2	0.6677	0.1858	0.4208	0.2259	0.6181	0.2070	0.2187	0.1588	0.2066	0.1382	0.4257	0.1712	0.6181	0.1300
3	0.2279	0.2291	0.1656	0.1701	0.2117	0.2123	0.1826	0.1851	0.1436	0.1549	0.1763	0.1787	0.2117	0.2128
4	-	-	0.1192	0.1195	-	-	-	-	0.0280	0.0278	0.1014	0.1029	-	-
5	-	-	-	-	0.1445	0.1471	0.0497	0.0449	0.0123	0.0112	-	-	0.1445	0.1786
6	0.0267	0.0252	-	-	0.0246	0.0254	0.0365	0.0378	0.0248	0.0267	-	-	0.0246	0.0354
7	0.0767	0.0751	-	-	-	-	0.0309	0.0317	0.0999	0.1054	-	-	-	-
8	-	-	0.0057	0.0061	-	-	0.1662	0.1683	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	0.0539	0.0574	0.1321	0.1410	-	-
10	-	-	0.2877	0.2907	-	-	-	-	0.0413	0.0409	0.1635	0.1639	-	-
11	-	0.0249	-	-	-	-	-	-	0.0144	0.0175	-	-	-	-
12	-	0.1792	-	-	-	-	-	-	0.0126	0.0119	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	0.0359
14	-	-	-	-	-	0.0184	-	-	-	-	-	-	-	0.0625
15	-	-	-	-	-	0.2779	-	-	-	-	-	-	-	0.1561
19	-	-	-	-	-	-	-	-	0.0637	0.0713	-	-	-	-
20	-	-	-	-	-	-	0.0154	0.0124	0.0050	0.0190	-	-	-	0.0031
21	-	-	-	0.0096	-	-	0.0747	0.0954	-	-	-	-	-	0.0103
22	-	-	-	-	-	-	0.0666	0.0659	-	-	-	-	-	0.0014
23	-	-	-	-	-	0.1109	0.0568	0.0549	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	0.0307	-	-
25	-	-	-	0.0697	-	-	0.1007	0.1016	-	-	-	0.2106	-	-
26	-	-	-	0.1074	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-	-	0.1727
34	-	0.0203	-	-	-	-	-	-	-	-	-	-	-	-
36	-	0.0844	-	-	-	-	-	-	-	-	-	-	-	-
38	-	0.0716	-	-	-	-	-	-	-	-	-	-	-	-
39	-	0.0652	-	-	-	-	-	-	0.2929	0.2946	-	-	-	-
40	-	0.0382	-	-	-	-	-	-	-	-	-	-	-	-
50	0.0010	0.0010	0.0010	0.0010	0.0011	0.0010	0.0012	0.0432	0.0010	0.0232	0.0010	0.0010	0.0011	0.0012

TABLE II

SNR WORKING POINTS OF THE CONSIDERED CODING SCHEMES FOR SEVERAL VALUES OF R_s AND R_B; THE VALUES OF $\frac{E_b}{N_0}$ ARE IN DB.

R_s	R_B	$\frac{E_b}{N_0}\Big\vert_{th}^{B}$	$\frac{E_b}{N_0}\Big\vert_{th}^{F}$	$\frac{E_b}{N_0}\Big\vert_{n_1}^{B}$	$\frac{E_b}{N_0}\Big\vert_{n_1}^{F}$	$\frac{E_b}{N_0}\Big\vert_{n_2}^{B}$	$\frac{E_b}{N_0}\Big\vert_{n_2}^{F}$
0.33	0.35	-0.14	-1.52	1.10	3.82	0.72	3.18
0.4	0.5	0.41	-0.52	1.00	0.76	0.78	0.44
0.45	0.5	0.42	-0.69	1.12	1.22	0.82	0.98
0.5	0.75	1.73	0.38	2.14	1.17	1.94	0.84
0.6	0.75	1.72	-0.14	2.12	0.98	1.97	0.63
0.7	0.75	1.75	-0.52	2.13	0.91	1.92	0.60
0.725	0.75	1.75	-0.69	2.18	2.11	1.96	1.59

codeword lengths. We have also proposed an optimization approach to design good codes for this context. Our results show that these codes are able to approach the ultimate performance limits even with relatively small block lengths.

REFERENCES

[1] A. Thangaraj, S. Dihidar, A. Calderbank, S. McLaughlin, and J.-M. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Trans. Inform. Theory*, vol. 53, no. 8, pp. 2933–2945, Aug. 2007.

[2] D. Klinc, J. Ha, S. McLaughlin, J. Barros, and B.-J. Kwak, "LDPC codes for the Gaussian wiretap channel," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 532–540, Sep. 2011.

[3] M. Baldi, M. Bianchi, and F. Chiaraluce, "Coding with scrambling, concatenation, and HARQ for the AWGN wire-tap channel: A security gap analysis," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 883–894, Jun. 2012.

[4] M. Baldi, F. Chiaraluce, N. Laurenti, S. Tomasin, and F. Renna, "Secrecy transmission on parallel channels: Theoretical limits and performance of practical codes," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 11, pp. 1765–1779, Nov. 2014.

[5] C. W. Wong, T. F. Wong, and J. M. Shea, "Secret-sharing LDPC codes for the BPSK-constrained Gaussian wiretap channel," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 551–564, Sep. 2011.

[6] ——, "LDPC code design for the BPSK-constrained Gaussian wiretap channel," in *Proc. IEEE GLOBECOM 2011 Workshops*, Houston, TX, Dec. 2011, pp. 898–902.

[7] A. Chakrabarti, A. de Baynast, A. Sabharwal, and B. Aazhang, "Low density parity check codes for the relay channel," *IEEE J. Select. Areas Commun.*, vol. 25, no. 2, pp. 280–291, Feb. 2007.

[8] J. Wang, S. Che, Y. Li, and J. Wang, "Optimal design of the joint network LDPC codes for half-duplex cooperative multi-access relay channel," in *Proc. 5th Int. Conf. on Intelligent Networking and Collaborative Systems (INCoS) 2013*, Xi'an, China, Sep. 2013, pp. 622–625.

[9] R. Khattak and S. Sandberg, "Jointly optimized rate-compatible UEP-LDPC codes for half-duplex co-operative relay networks," *EURASIP J. Wireless Commun. and Networking*, vol. 2014, no. 1, 2014.

[10] M. Bloch and J. Barros, *Physical-Layer Security*. Cambridge University Press, 2011.

[11] A. D. Wyner, "The wire-tap channel," *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975.

[12] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[13] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[14] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. IEEE GLOBECOM 2001*, San Antonio, Texas, Nov. 2001, pp. 995–1001.